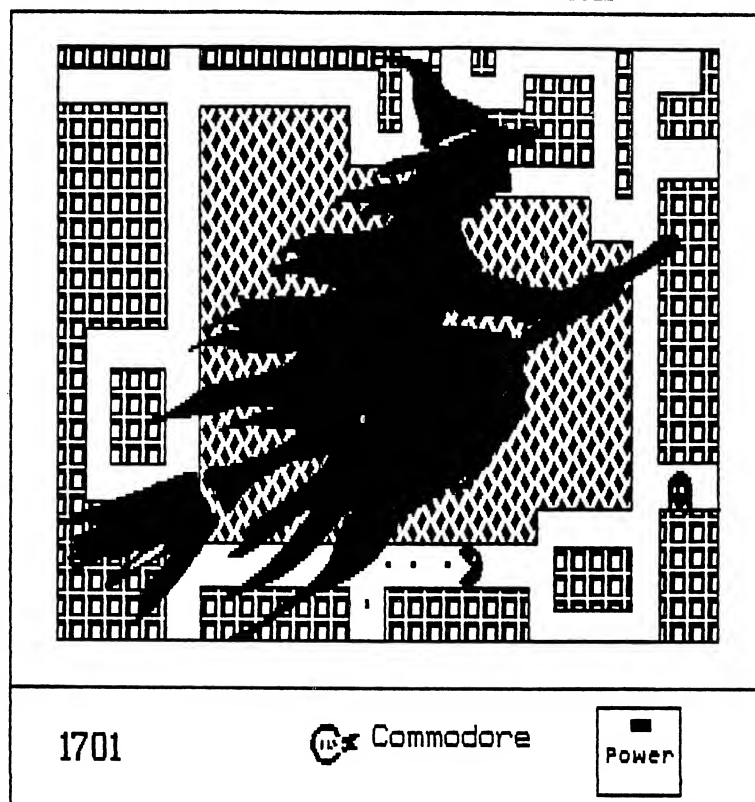




C.H.G.S.



"OCTOBER NIGHTMARE"



MONITOR



1988

Vol.3 #9



OBLIGATORY STUFF

PRESIDENT	Richard Maze	586-3291
VICE-PRESIDENT	Barry Bircher	359-1925
TREASURER	Harry Chong	
LIBRARIAN	Earl Brown	543-2068
ASS'T LIBRARIAN	Steve Bogues	
EDITOR	Ken Danylczyk	545-0644
ASS'T EDITOR	Greg Rezansoff	
MEMBER AT LARGE	Gord Williams	

THE MONITOR is published monthly by the COMMODORE USERS' GROUP OF SASKATCHEWAN (CUGS), Regina, Sask., Canada. CUGS meetings are held at 7 pm the first Wednesday of every month (unless otherwise noted) in the North-West Leisure Centre, corner of Rochdale Boulevard and Arnason Street.

Anyone interested in computing, especially on the C64, 128 or 64C, is welcome to attend any meeting. Out of town members are also welcome, but may be charged a small (\$5.00) mailing fee for newsletters. Members are encouraged to submit public domain software for inclusion in the CUGS DISK LIBRARY. These programs are made available to members. Any member is entitled to purchase DISKS from our public domain library for a nominal fee. Programs are 'freeware', from computer magazines, or the public domain. Individual members are responsible for deleting any program that he/she is not entitled to by law (you must be the owner of the magazine in which a particular program was printed). To the best of our knowledge, all such programs are identified in their listings. Please let us know if you find otherwise. Contact Earl Brown, 737 Rink Ave.

CUGS is a non-profit organization comprised of C64, 64C, C128, and 128D users interested in sharing ideas, programs, knowledge, problems and solutions with each other. The more members participate, the better the variety of benefits. Membership dues are pro-rated, based on a January to December year.

In this issue:

COME ONE! COMAL!	- an old "new" language
MEETING PLACE	- Date, Time, Place, Agenda
EDITORIAL	- On being active!
READY?! OR! NOT!	- Bircher being logical
SIR RICHARD'S BASIC	- Helping Canada Post
SCRATCH 'N' SAVE	- Brown new duties!
128 WINDOW	- new ROM ruminations
MORE MIRTH THIS MONTH-	Don't worry ... be happy

MEETING PLACE

AGENDA:

RICHARD MAZE - PRESIDENT'S COMMENTS

THE GREAT CANADIAN COMPUTER CLUB OFFICIAL
1988 PRE-CHRISTMAS BOOK REVIEW AND ADVICE
PRESENTATION

*****coffee****visiting****disk-picking*****

A QUICK COMAL OVERVIEW
(by Ken Danylczyk)

EDITORIAL:

Making a start.

Ever had a dream, or a "pie in the sky" plan? Ever wondered what WOULD have happened if...? Life is full of beginnings, doors slightly ajar, asking to be opened and stepped through. But you've got to have MORE than a plan, MORE than a dream, to see what's behind any one of those doors. Someone's got to give the door a shove, make an effort and walk on through!

AWRIGHT! Enough philosophy already! So what's this all got to do with computers and computing?

It's been 9 issues since this last occurred (yep, I counted!), so, this being the month just before our executive elections, I thought it might be wise to harangue you once again!

Last issue I promised more about BASIC programming or programming in general. Well, that'll be part of this, too!

WE NEED CLUB MEMBERS TO BE ACTIVELY INVOLVED IN THIS CLUB AND ITS FUNCTIONS. I could quote platitudes forever - 'Nothing ventured, nothing gained' - 'Every fire starts with a spark' - 'mighty oaks from little acorns grow' - 'floats dirt and stain right down the dr...' ... oops!

What I began to say last issue (and will continue saying for the next while) is that each of us reaches a point in our understanding of a skill where we 'plateau' - we level off and cease growth for a while. There's nothing wrong with that - it CAN be good - give us time to reflect. But it IS a critical point in our development. If we allow ourselves too long at the 'plateau', we become complacent and casual, believing we have exhausted our abilities. We begin to think of ourselves as limited and dependent on others for any improvement. The 'doors of opportunity' present themselves regularly, at every turn, but it gets easier each time to turn from the door, thinking ourselves ill-equipped or too unskilled to open and enter any of the doors.

Sorry, them metaphors keep sneakin' up on me. What I'm trying to say is that every one of us has a skill or talent of SOME sort to offer the club. Don't YOU be the judge of what you might do for your fellow computerists; pull yourself off the plateau - open a door - take just ONE step through - offer what you're able to your club - serve some office, write a review or article, offer to make some presentation. Just like staying ON the plateau gets easier as time passes, continuing onward and upward gets easier with every door you open. Try it - you'll find it's true.

And what's all that got to do with becoming a programmer? Well, USING programs makes a computer seem worthwhile, but changing or (better yet) CREATING a program to suit YOUR needs makes the USER FEEL WORTHWHILE. Skills? Start with what you have - and BUILD - a step at a time, slowly, gets you to and through those doors. AND with each door you open, you move up from your plateau with NEW SKILLS, which make the next door easier!



(tum tum tumteetum tum - for proper effect have your husband or wife softly hum "LAND OF HOPE AND GLORY" while you read the preceding paragraph)

And THAT GOES DOUBLE FOR PROGRAMMING! I know you own programs that others wrote that do everything you want ... sort of ... and it IS much easier to use a "ready-made" program (Hi ho hi ho, we sit on our plateau). I'm going to take the next editorial or two to present you all with one more door, a chance to move off your USER ONLY plateau and, however briefly, dabble in the realm of PROGRAMMER.

Let's try for that first step off the plateau (the first step really IS the hardest)! Just to help us all try that first step with an equal stride, we'll make that step a 'dabble' with a new language - COMAL! Why COMAL? - Why not?!

Seriously, folks, COMAL (COMMon Algorithmic Language) resembles BASIC but incorporates features touted by the "sophisticated" languages like Pascal and C. It allows someone with simple BASIC knowledge to experiment with STRUCTURED PROGRAMMING, and TURTLE GRAPHICS, or advanced SPRITE AND SOUND CONTROL. Best of all, it comes FREE (almost) on each COMAL club disk. It's available for MS-DOS, Commodore, Unix, and CP/M machines, and each operating system uses a common CORE of commands and structure, making the programs completely transportable. An article in this issue, and one of this month's presentations will guide you gently into some simple COMAL experiments, but, if you'd rather stay with BASIC, we'll include that, too. The objective is to get you to give it a try, open the door, step through, even for a little while. You MIGHT like it!

Wrichard Writes...

A Message from Maze!



October marks a turning point of sorts. The kids are back at school and we are starting to get back to the schedule the summer disrupted. At CUGS, it is also our second meeting after the summer and the first month of operation of our bulletin board. This month's meeting presentations are two that have not been tried before. Everyone has probably bought or received as a gift a computer book of some sort or other. We've seen other books that we are not sure of. At this meeting we are going to explore books - both the good and the bad. Hopefully, if there is a book you were thinking of getting, you can now have somebody else's opinions of the book to help you decide if it's worthwhile for you. We are also going to have a presentation on COMAL which should be of particular interest to anyone who would like to use a language other than BASIC for writing programs. Our library has a number of COMAL disks which makes it a little easier to use a new language (when there are programs already available).

The bulletin board has enjoyed some success, but to be really successful and meaningful, it requires the cooperation of all members. This doesn't just mean calling the board, although that is a good start. It means leaving messages, giving your opinions, uploading files as well as downloading. It's all backed up daily. You can't wreck it. You CAN make it very worthwhile and enjoyable for all. The direction the board takes will be completely up to you the members of CUGS because you are the only ones allowed on.

While I am blowing horns about involvement, it is also time to mention that CUGS elections are coming up next month. If any of you would like to help determine the direction that CUGS is going, leave your name with EARL who is acting as returning officer for this election. The more the merrier -- and the more ideas, the stronger the direction CUGS can take.

Scratch 'n' Save:

by Earl Brown

Fall is election time. Not just civic and federal, but here at CUGS as well. All those executive members who are planning to run once more, please let me know, as this job has been assigned to me this year. As for the rest of you, may I suggest you think about offering your time in one of the executive positions. It is not necessary to know very much about programming a computer, or, for that matter, know very much about a computer whatsoever, but the desire to organize, share, find out, collect, write, communicate, discuss, handle money. Any willingness to help with one of these items could make you a good candidate for one of the positions on CUGS executive. Don't just think about it - give me your announcement TONIGHT!

In the last issue of MONITOR, Ken acknowledged my name as the author of the '1571 FIX ROM' article. This is not so. All I did is key in the information via POCKET WRITER 2 for those of you interested. The original material was supplied in printed form from COMMODORE as was the update material on C128 ROMS to be published when Ken finds the room in our newsletter.

Barry Burgess brought to my attention a bad file on DISK 7 of our 128 PROGRAMS. The bad file was the first program on the disk, entitled "MULTITERM128 3.6". If any of you purchased this disk from the library, I would be more than happy to replace it, as the file in question has since been fixed. By the way, if any of you do come across bad files or disks at any time from our library, please let me know (preferably on a piece of paper or by phone) so that the file in question can be corrected or deleted as soon as possible. I thank you!

Three more 64 disks have been added to our library this month. They are COMMUNICATIONS 7, COMMUNICATIONS 9, and SOUND 13. Don't forget to include this month's listings with your catalogue. They're being deliberately printed on the last page of the newsletter for easier removal for your files!



128 WINDOWS

for REU and REU Users, Too?

Ram Expansion Data Storage on the C-128

by Shaun Hase

I was goofing around with my computer one morning, when I should have been studying, and I developed a useful routine that could be used in a variety of situations. I wrote this program due to the fact that I would like to write a bulletin board system. One of the many things holding me back, besides frequent failed attempts at it in the past, is that running a bulletin board wreaks havoc on your system, especially your drive.

I thought that if I could use the ram expansion as a type of data file for bulletins, messages, menus, etc..., it would save wear and tear on the drive. The only draw back to using a system like this is the power dependence of the REU - without power it doesn't retain memory.

I began to think about how I could STASH and FETCH data, piece by piece, to the REU. You can't just save the data, it has to be stored in some memory location, and, since the variable bank is unstable at the best of times, another way of storing the data had to be figured out. I pulled out a memory map and looked for a location to store the data, which is where the number 786 came from. It is an unused memory location. Perfect.

So, a sequential file is opened, and the data is retrieved one byte at a time. The numerical value of the byte is POKed into 786, and then that lone byte is STASHed to the memory expansion. The data is retrieved just as one would retrieve data from the disk to print to the screen, printer, etc.... As the data is brought in, a counter is incremented to keep track of the relative size of the file. After the file is in the REU, it can be accessed by a simple FOR/FETCH/NEXT loop. It's not much faster than a disk access, since it can only run in SLOW mode, but it saves the drive from running.

So, how could this be used? It could be used to store a variety of sequential files, with lengths and locations in the REU stored in arrays, that could be called up through the quick FETCH routine. Drive accessing would be cut down to a minimum, since the data need only be loaded in once. The data in the REU won't be erased, even on a soft restore (RESET button), unless the power is turned off or the data is overwritten by other information.

```
10 c=0
20 input"Sequential file name";fi$
30 open8,8,8,(fi$)+"$,r"
40 get#8,a$
50 ifst=0then60:elseclose8:goto100
60 poke786,asc(a$)
70 stash1,786,c,0
80 c=c+1:printa$;
90 goto40
100 print:print"Press any key for Ram Disk Read"
110 getkeya$
120 fort=0toc
130 fetch1,786,t,0
140 printchr$(peek(786));
150 next
```



128 ROM UPDATES:

by Earl Brown

(Editor's Note: Earl recently upgraded ROMS in his 128 and 1571 equipment, and began with the last issue to explain some of the reasons for the upgrade installation. This article continues with the C128 ROMS.)

NEW COMMODORE C128 OPERATING ROMS

Part Number 318018-04 --> BASIC LOW (\$4000-\$7FFF - replaces Part Number 318018-02)

1. LIST and DELETE commands - Previously, these did not report as errors certain non-numeric characters passed as arguments, (eg. 'LIST A'). This has been corrected by adjusting an erroneous relative branch in the 'RANGE' subroutine.
2. CIRCLE command - Previously, an unspecified Y-radius defaulted to the X-radius (as it should), but the X-radius value had already been scaled for the X-axis and not the Y-axis. This has been corrected by scaling the radii AFTER the defaults have been established.
3. RS-232 Status - Previously, accessing ST after RS-232 I/O resulted in an incorrect Status being returned from, and a zero written to, location \$10A14, possibly corrupting the BASIC variable area. This was a result of BASIC calling the Kernal routine 'READSS' with the incorrect RAM bank in context. This has been corrected by substituting the correct BASIC subroutine call.
4. CHAR command - Previously, using CHAR with the 80-column text screen (GRAPHIC mode 5) resulted in RAM corruption at locations \$D600 and \$D601 of RAM bank 0 (the BASIC text bank) due to BASIC calling the Editor PLOT routine without the I/O block in context. This has been corrected utilizing two patch subroutines.
5. RENUMBER command - Previously, the pass 2 routine, which was to pre-scan BASIC text and report 'out of memory' errors prior to actually changing anything, was seriously flawed. This has been corrected utilizing a patch subroutine.
6. DELETE command - Previously did not limit-check itself when moving down BASIC text, therefore, it was possible to crash when DELETEing lines at or near the top of memory, near the MMU configuration registers. This has been corrected utilizing a patch subroutine. Also, DELETE previously exited to MAIN via 'JMP', effectively ending the evaluation of the current command string. This has been corrected by substituting an 'RTS', allowing direct commands like 'DELETE 10:PRINT"DELETED LINE 10"' to work correctly.
7. PLAY command - Previously, the SID frequency tables were not exactly NTSC concert pitch. Also, there was no provision for adjusting the frequency for PAL systems. This has been corrected by changing the (NTSC) frequency tables, creating new PAL tables, and utilizing patch code to select from the appropriate table as determined by the Kernel PAL NTSC flag.
8. The BASIC ERROR handler previously failed to clear pending string temporaries when an error was TRAPPED. This has been corrected via patch code to reset TEMPT to TEMPST.
9. The powerup notice has been updated to 1986, which will serve as an immediate visual indication of the ROM update status. Also, a new notice has been placed at \$7FC0.

10. The ROM signature at location \$7FFC and \$7FFD (lo/hi) is \$8DEF (new since last release).
11. The ROM revision byte at location \$7FFE has been incremented from \$00 to \$01.
12. The ROM checksum byte at location \$7FFF, has changed from \$4C to \$61.

Part No. 318019-04 -> BASIC HIGH, MONITOR
(\$8000-\$BFFF - replaces Part Number 318019-02)

13. RSPRITE and RSPPOS functions - Previously, they accepted as parameters sprite numbers in the range 1-16, which is incorrect. This has been corrected by limiting the range check to 1-8, and reporting an illegal quantity error for sprite numbers outside this range.
14. PRINT USING command - Previously, there was an anomaly involving the use of floating dollar symbols (\$) and commas. The command 'PRINT USING "#,##\$.##"; 123.45', for example, resulted in the output '\$,123.45', which is incorrect. This has been fixed utilizing a patch subroutine which checks specifically for the '\$,' occurrence and substitutes a '\$' (' ' = fill character) whenever found.
15. The relative coordinates for all graphic commands (except MOVSPR) were incorrectly processed. The problem was apparent when negative relative coordinates were used, which resulted in an illegal quantity error. This has been corrected by substituting a different subroutine call to pre-existing code. This change affects the BASIC commands LOCATE, DRAW, PAINT, BOX, CIRCLE, GSHAPE, and SSHAPE. This change also allows negative absolute coordinates to be accepted (previously they resulted in an illegal quantity error), although the legal range remains an 8-bit value: 0-65535 (unsigned) or -32768 to 32767 ('signed' means -1 is equivalent to 65535).
16. DOPEN and APPEND commands - Previously, it was possible to open 2 or more disk channels with the same logical file number without incurring an error report. This has been corrected.
17. MATH package - An original bug fix (ref: double zero bug) to the (F)MULT routine has been found to result in small errors (such as 25 = 32768.0001). This has been corrected by fixing the original (dbl-0) problem in a different way.
18. A copyright notice has been placed, starting at \$BFC0.
19. The ROM signature at location \$BFFC and \$BFFD (lo/hi) is \$CDC8. (new since last release).
20. The ROM revision byte at location \$BFFE has incremented from \$00 to \$01.
21. The ROM checksum byte at location \$BFFF, has changed from \$3A to \$C5.

Part No. 318020-05 > EDITOR, KERNAL, CP/M
(\$C000-\$FFFF - replaces Part Number 318020-03)

22. CAPS LOCK Q - Previously an error in a key matrix decode table caused a lower-case 'Q' to be passed when the keyboard was in CAPS LOCK mode. The table has been corrected by substituting the correct value for upper-case 'Q'.

23. FUNCTION KEYS - Previously, the function key handler, part of the SCNKEY routine at CKIT2, failed to detect a function key string pending. This has been corrected via a patch routine, which will ignore new function key depressions until the string in progress has been output (i.e., KYNDX = 0). Also, DOPFKY now exits via SCNRTS, instead of simply RTSing.
24. IOINIT system initialization - Previously, the RS-232 pseudo-6551 registers were not initialized because these values are expected to be given by the user whenever RS-232 channels are OPENed. Apparently many C64 users have taken advantage of the fact the C64 'happened' to clear these locations and fail to specify critical parameters. These RS-232 registers are now initialized to default to: no parity, full duplex, 3-line, 1-stop bit, 8-bit words and 300 baud, via a patch subroutine.
25. IOINIT PAL system initialization - Adjustments have been made to the 8563 initialization values for PAL systems. The PAL horizontal total (register 0) changes from \$7E to \$7F. The PAL vertical total (register 4) changes from \$27 to \$26. These changes shift the cycle time from 20.320us to 20.032us. The patch required a patch subroutine, as well as a change to VDCTBL.
26. BASIN system call - Previously, attempting input from a logical channel to the screen (e.g., via INPUT#) resulted in LINE TOO LONG errors. This has been corrected utilizing a subroutine patch to preserve bit 7 of CRSW, which serves as a flag to the Editor that a (pseudo) END-OF-LINE has been reached. Also, TBLX is copied to LINTMP to correctly locate the current cursor line for the Editor. Please note that switching between the 40 and 80-column text screens, opening and closing windows, or clearing text screens can confuse logical screen channels. The Editor variable LINTMP (\$A30) is a global, not local, variable as it should have been. Users can POKE LINTMP with the logical screen line number before INPUT#'s as a work-around.
27. OPEN RS-232 system call - Previously, it was possible to receive a carry-set status, normally indicating an error, when no error existed after OPENing an RS-232 channel. This has been corrected totally in-line by a modification to the code which checks for the proper X-line hardware status.
28. LOAD system call - Previously the normal (a.k.a. SLOW) load mechanism did not preserve the starting address of any LOADs, which made the BASIC 'BOOT "file"' command form malfunction unpredictably. This is apparent only when used with 1541 drives. This has been corrected via a patch subroutine which saves the starting address of all LOADED files at SAL and SAH, the same place the fast (a.k.a. BURST) load mechanism does.
29. DMA system call - Previously, the Kernel forced the I/O block into the user's memory configuration at all times, which is no longer necessary and, in fact, seriously limits the functionality of the RAM expansion cartridge. This has been corrected by a ROM patch routine, which affects all Kernel DMA system calls, as well as the BASIC FETCH, STASH, and SWAP commands. Also, previously, it was possible for an IRQ to occur between the 'arm DMA' and 'trigger DMA' sequences, resulting in a DMA operation with the system configuration in context regardless of desired configuration. This has been corrected by adding 'PHP/SEL...PLP' instructions around the JSR to DMA RAM code at \$3F0. Applications using the DMA RAM code at \$3F0 should do likewise. Finally, in this patch changes were made to enable DMA operations to all RAM banks by correctly using the VIC bank pointer found in the MMU RAM configuration register (\$D506, VA16-bit-6 and VA17-bit-7). Applications using the Kernel routine at \$FF50 will inherit these changes automatically. Please note that MMI interrupts can screw-up DMA operations, as they cannot be masked.



30. A copyright notice has been placed, starting at \$CFCO.
31. The ROM location \$CFF8 is reserved for national character ROM checksums. This does not apply to US ROMs, which contain \$FF here. (new since last release).
32. The ROM location \$CFF9 is now reserved for country codes. The US ROMs contain \$FF here.
33. The ROM location \$CFFA and \$CFFB (lo/hi) contain the national character set signature. This does not apply to US ROMs, which contain \$FFFF here. (new since last release).
34. The ROM signature at location \$CFFC and \$CFFD (lo/hi) is \$8F76. (new since last release).
35. The ROM revision byte at location \$CFFE, has incremented from \$00 to \$01.
36. The ROM checksum byte at location \$CFFF, has changed from \$C3 to \$3C.
37. The Kernel revision byte at location \$FF80 has incremented from \$00 to \$01.

Sir Richard's BASIC 2. the ENVELOPE please!

In the last article, I examined setting up a title page using the word processor. This article continues the series. My word processing program is Paperclip. If you are using a different word processing program, you may find you will have to make some slight changes in the formatting.

After a letter has been prepared, the next step is to prepare the envelope to mail it. Often this involves writing directly on the envelope, which doesn't look very professional, or preparing a mail label, which involves changing the printer margins and often loading and running a separate program. In this article I will show you how to print directly on the envelope. Your envelope file can then be saved with the letter file. Once you have prepared one envelope file, it is a very simple matter to modify that file for a different address. In fact, you could save every envelope file you use separately so it just has to be reloaded to use again, or you could just save a master envelope and load and modify it when needed. The only restriction is that you must have a printer able to take single sheets.

This article is based on using a number 10 envelope (common business size). For other sizes there are only 2 changes required which I will point out to you at the time.

The first step is to give the formatting commands which tell Paperclip the paper size and where to set the margins. To permit maximum flexibility in envelope size I simply set the paper size to 11 inch or 66 lines (pp66) and indicate I will use 60 of these lines (pg60). The margins are set at 2 and 75 (lm2:rm75). The left margin setting of 2 is important so that the return address is close to the left margin. The right margin setting just has to be large enough to handle very long addresses. 75 will be 1 1/2 inches from the right edge of a #10 envelope. It is also a good idea to come down 1 line (vpl) from the top of the envelope before beginning printing. This is just in case the top of the envelope is not properly lined up in the printer. Omitting this could result in the tops of the first line letters being cut off. All these formatting commands can be put on one line which would look like:

```
<checkmark> pp66:pg60:lm2:rm75:vpl
```

Now it is time to enter the return address. This usually consists of four lines: name, street, city & province, and postal code. If sending to another country then Canada is included before the postal code on the postal code line. Just type these in with a return at the end of each line. For example:

Richard Maze
143 Birchwood Cres.
Regina, Sask.
Canada S4S 5S3

To set up the destination address on the envelope there are two formatting commands that must now be set. These commands are the ones that will also be changed for different sizes of envelopes. The commands change the left margin to 40 (lm40) and go down 6 blank lines (ln6). Depending on the difference in size of the other envelope from a #10 envelope will determine how these must be changed. If you don't like the look of the envelope, change these commands. This formatting line will appear as:

```
<checkmark> lm40:ln6
```

The destination address may consist of 4 or 5 lines. The extra line appears if you are sending to a section of a large institution. The other lines are the same as for the return address. If the fifth line is not required just omit it completely. With margins of 40 and 75 you have up to 35 characters for each line which should be more than sufficient. An example is:

CUGS
143 Birchwood Cres.
REGINA, Sask.
S4S 5S3

In most cases, your envelope is finished. The advancing of the paper to the end of the page will roll your envelope out of the printer so no more commands are required. Occasionally, you must indicate that your letter is to go to a particular person within a department. If you can't address the envelope directly to that person you can indicate with: attention:<name> in the bottom left corner of the envelope. To include this on your envelope requires two more lines. The first line changes the left margin to 10 (lm10) and goes down one blank line (ln1). The second line contains the word 'attention:' followed by the person's name. These are usually all in capitals.

Overall, an envelope would contain the following commands and text. (Note: substitute <checkmark> [english pound sign] for %).

```
%pp66:pg60:lm2:rm75:vpl
Richard Maze
143 Birchwood Cres.
Regina, Sask.
Canada S4S 5S3
%lm40:ln6
CUGS
143 Birchwood Cres.
REGINA, Sask.
S4S 5S3
%lm10:ln1
ATTENTION:XXXXXXXX
```



When setting up your printer, make sure the top of the envelope is lined up properly at the start of a page and that the left margin setting particularly is set properly. To print out the envelope, simply use <control><shift><o> which will print out one page with default settings.

The Logical Bircher: Or is that And, Or...

by Barry Bircher

AND OR WHAT? Part II by B. Bircher

When does 1 or 3 = 3
 64 or 1 = 65
 8 or 8 = 8
 8 or 16 = 24

When your compu.....you read last month's article didn't you?

In last month's "AND" column I discussed the effects of ANDing bits which allow you to mask off certain bits. This week I want to call on the "OR" effects on bits. If you understood ANDing then you will find ORing even easier. The truth table (a fancy name for all possible outcomes on one table) for the OR effect is as follows:

BIT form			
1	1	0	0
ORed to 1	ORed to 0	ORed to 1	ORed to 0
= 1	1	1	0

If either bit X "OR" bit Y is ON (1) then the output is also ON (1). Both have to be OFF (0) in order for the output to be OFF (0).

What possible use is ORing, you may well ask. ORing allows you to turn on individual bits ON (1) without affecting the others. If the corresponding bit is already ON (1) then it will remain ON (1).

eg:

Binary	Decimal
10011101	157
ORed 11010011	ORed 211
= 11011111	223

Using last month's example of the sprite register, We may need to turn on sprites 8, 7. As you recall sprite 8 is the highest bit with a weighted value of 128 decimal and sprite 7 is 64. We would then OR location 53269 with 192 (128+64). In binary it looks like this;

01001001	(this could be any value)	73
11000000		192
= 11001001		201

This turns on bits 7 and 8 regardless if it was already on or not.

Exclusive OR

A variation on the OR effect is the Exclusive OR or EOR for short. It has the similar effect as OR but with a twist. If both bits being ORed are ON (1) then the output is 0/off. eg;

1001101
EOR 1011110
= 0010011

The effect of EOR is to flip the bits, if they were ON (1) then now they are OFF (0). (this effect is seen only on the corresponding bits that are set in the EOR byte.) There is, unfortunately, no EOR statement in BASIC.

^{NOT}
NOT effects ON bits is simple. If it is ON (1) then its 'NOT' is OFF (0).

1 0 1001110
NOT = 0 NOT = 1 NOT = 0110001

The net result of ANDing, ORing and NOTing is individual control of bits in a byte. The effects of these operators is not clearly seen when looked at in decimal. When these bytes are looked at in there native environment and weighted (valued at 128, 64, 32, 16, 8, 4, 2, 1 respectively) then the results are more easily understood. To a BASIC programmer, these can be and are usable to their advantage. These operators can be used in an equation (Boolean Algebra) that makes digital circuitry as easy to read and assemble as making up an equation, no matter how complex the job at hand... but that's another chapter. See you at the meeting.

NEXT MEETING

CUGS MEETING WEDNESDAY - NOVEMBER 2, 1988

NorthWest Leisure Centre Room #1
7:00 pm

AGENDA

1. Introduction
2. Election of executive for 1989
3. Software Preview
4. Prize draw



At the CUGS meeting in November, we will have the election of officers for 1989. The executive positions consist of:

President
Vice-President
Secretary/Treasurer
Editor
Assistant Editor
Librarian
Assistant Librarian
2 members at large

The duties of the executive members primarily consist of:

- planning meeting agendas.
- looking after the disk library
- publishing the 'Monitor'

If you would like to help the club by providing your expertise as an executive member, we would be more than happy to accomodate you. New, fresh ideas are not just welcome, they are a necessity to the continued success of our club. So, please, get involved. EARL BROWN has agreed to act as a nominating committee (of one). Give your name to Earl and become a CUGS executive member.

Through the generosity of Bart, at Software Supermarket, we will be once again giving club members a chance to preview what's new in software before Christmas. Last year we held this in December and, although it was well received, many felt it was too late. Bart feels that his new Christmas stock should be available by November 2, so we will give you a chance to have a hands-on with the newest, latest software.

We will have a number of C64's and 128's set-up so you can examine whatever you want.

Mark it on your calendar now - WEDNESDAY, NOVEMBER 2.

TIME MACHINE II

A Known History
An Uncertain Present
A Possible Future
Of the Micro Computer Industry
As Seen by Barry Bircher

Continuing from where we left off last month, I want to lay out some important dates in computer history. The computer industry is growing fast. Below you will find a quick reference to how fast things got going.

- * 1634 :The first true mechanical calculator is invented by Wilhelm Schickard.
- * 1642 :Pascal invents his adding machine.
- * 1674 :Leibnitz's calculating machine constructed.
- * 1786 :Babbage starts on model of Difference Engine.
- * 1834 :Babbage conceives idea of Analytical Engine (first concept of modern-day computer).
- * 1937 :Atanasoff conceives idea of the "ABC" machine.
- * 1942 :ABC near operational but incomplete due to war.
- * 1943 :Harvard Mark I operational.
:Construction starts on ENIAC.
- * 1950 :SWAC operational.
- * 1951 :First UNIVAC is delivered to U.S. Census Bureau.
:The first Ferranti Mark I version delivered to Manchester University.
:LEO I operational.
:IAS I operational.
- * 1952 :MANIAC and ORDVAC copies of IAS are operational.
:IBM 701 delivered.
:EDVAC finally finished.
- * 1955 :ENIAC shut off for last time.
- * 1959 :Harvard Mark I shut down for last time.
- * 1960 :UNIVAC LARC delivered.
- * 1962 :ATLAS installed.
- * 1971 :Intel corp. introduces the 4004 CPU, the first "microprocessor". Shortly afterwards, Atari ships out "PONG" the first video game.
- * 1972 :Intel Introduces the 8008 CPU
- * 1973 :The term "Microcomputer" first appeared in print in reference to the "Micral" (Intel's 8008) introduced in non kit form in 1973. In 1973 a guy by the name of Gary Kildall built a computer in his basement and developed CP/M (Control program/monitor) and later marketed it as CP/M 1.4.

It was about 1974-75 that things really got going for the home computerist.
- * 1974 :Motorola introduces the 6800 chip. Creative Computing was the first Home computer users magazine. Intel releases the 8080 chip, an early industry standard, was used in the MITS Altair 8800. An article in Popular electronics on the Altair 8800 was the first machine to be called a "personal computer".

- * 1975 :MOS Technology creates the 6502 CPU the one and the same used in our adorable Commodore-64. MOS Technology was formed by some of the Motorola 6800 CPU designers who left and formed their own company. Motorola sued but didn't go to court. However they did manage to hurt MOS and were thus acquired by Commodore.
- * 1976 :Zilog introduces it's Z80 CPU, this one was one of the more popular ones for several years. Zilog was created by 3 Men, Frederico Fraggin, Ralph Ungermann and Matsushita Shima.
:Digital Research copyrights CP/M; Gary Kildall founder and president.
:The first Apple computers is sold in stores by Steve Jobs and Steve Wozniak.
- * 1977 :Radio Shack Trs-80 is unveiled.
:Commodore PET is introduced.
:The Apple II debuts.
- * 1978 :Apple introduces the II Plus.
:Atari announces its models 400 and 800. Epson puts out the MX-80 dot matrix printer.
:Intel releases the 8086 16 bit CPU.
- * 1979 :Micronet goes online. Was later changed to CompuServe one year later.
:The source goes online.
:Motorola brings out the 68000 16-bit microprocessor first used in the Apple Macintosh.
:Intel 8088 CPU is introduced which is still widely used today in IBM's PC's.
:Wordstar from MicroPro is sold and becomes very popular.
:VisiCalc by Software Arts is released.
- * 1980 :The computer game ZORK is sucked off of the mainframe computer at M.I.T. and put into home computers.
:Sinclair shows it's ZX80 chip used in a computer marketed by Timex, later to be called Timex-Sinclair.
:The Apple III shows up.
:TRS Model III and CO-CO appears.
- * 1981 :Commodore unveils the VIC-20.
:National Semiconductor shows the 32000 32 bit chip. Epson introduces the HX-20 Laptop computer. Ashton-Tate ships dBase II, now an industry standard.
:Adam Osborne introduces the Osborne I (CP/M)
:IBM PC is introduced.
- * 1982 :Commodore unveils the C-64 ("I Adore my 64"), the Max Machine, the BX256 16 bit professional, the PI28, and the BI28 professional
:The First IBM clone appears.
:Lotus 1-2-3 appears.
- * 1983 :Apple unveils Lisa and the IIfx.
:IBM PC XT and PCjr. are announced.
:Radio Shacks announces the model 100 laptop, the TRS-80 model 4 and the Tandy 2000.
- * 1984 :Jack Tramiel, the founder and president of Commodore leaves the company.
:Apple shows the Macintosh and IIfx
:Commodore purchases Amiga Corporation.
:IBM PC AT appears.
:Tandy 1000 and 1200 debut.
- * 1985 :Commodore unveils the C-128 and the AMIGA 1000.
:Atari XE and 520ST debut.
:Intel shows up with the 80386.
:IBM introduces the PS/2 line of computers. (Intel 80286, 80386 CPU)
- * 1986 :Apple introduces the Mac Plus and IIfx.
:Berkeley Softworks ships out GEOS.

- * 1987 :Commodore unveils the Amiga 500 and 2000.
:Berkeley Softworks ships out GEOS 128.
- * 1988 (-to present) : Commodore introduces 2 new Amigas, the Amiga 2500AT (68020 AND a 80286 CPU), and the Amiga 2500UX (UNIX OS).

Next month, I'll take you to what I see in the computers future.

See you then...

FUTURE MEETINGS:

CUGS MEETINGS OCTOBER - DECEMBER 1988

October 5
November 2
December 7



Meeting times are 7:00 - 9:00 pm

All meetings are held at the NorthWest Leisure Centre (Room #1)

No BS BBS!

(BS=Bad Slang, BBS=Bulletin Board System)

COR CUGS GETS WIRED

The CUGS BBS is now operational. In the past month, a few members have called the BBS.

If you haven't called yet, do so. The number is 586-3291. In the remainder of this article, I will point out a few general features of the BBS and some quirks which some of the early callers found by trial & error.

When you call, you can just press RETURN and, by answering questions, will get yourself registered. When you enter the phone number, enter it in the form: 306-XXX-XXXX Your phone number is one of your identifiers and requires the area code as a prefix. You are allowed a handle and all reference to you on the board is by your handle. If you don't want to use a handle (it is OK as long as everyone knows it) enter your name as your handle. You will also be asked for your password which is your other major identifier on the board.

The first time on the BBS, you won't be allowed to do very much. You may want to examine the main menu items (press ? at menu to get a word list and/or press h to get a 3-4 line description of each menu area).

Once you have called and registered, your status will be upgraded so you can properly use the board. At most, this will take a day, and if I am around, it will be done while you are on-line. When you call back, enter your phone number, and password at the prompts. If the program won't accept your password, try entering it again in capitals. For some reason, it seems that the password gets saved in uppercase.

There are 6 Bulletin areas available. These are Public, BBS numbers, For Sale, CUGS, BBS Comments, and Computer Tips. Each of them has an introductory bulletin outlining the general intended use of each bulletin area. Feel free to browse through these bulletins. Pressing ? will give you a menu of options within the bulletins area. Don't forget to post a bulletin to some else in the Public area at least.

E-Mail is private mail. If you want to send a private note to another registered user, (press L from the main menu to get a list of registered users), do it here.

The files area is accessed by entering either P (Punter transfer) or X (Xmodem transfer). Both of these allow uploading and downloading. Until we get a better indication of which way the board is going, the files area will not be developed very extensively. If you want to have a more extensive files area, let me know. The capability exists for a large partitioned files area.

You can enter the board either with a Graphics or an ASCII terminal program. In graphics, you will be able to use colors, etc. in the bulletins and when sending mail. To check your status (press Y from main menu) you first must be in ASCII. You can change between ASCII and Graphics by pressing Z which toggles between the two so if you are using a Graphics terminal you can easily switch to ASCII and back. If you are using an ASCII term, you may find that you get some garbage characters at certain times. These are caused by the translation of color codes to ASCII. It simply means that what you are observing was posted in color(s).

When posting bulletins or sending E-mail there are a couple of things that you should know. First, use one or two shifted spaces to send a blank line in text. If you don't, the message will only be saved up to the first blank line encountered. Also, the editing function has a quirk in it. Read the instruction file for info on how to handle this..

The BBS is like any other club function. Its value will be determined completely by the use that you, the members, make of it. Please also leave comments either in the bulletin area - BB Comments - or as mail to the sysop. Thanks.

COME ONE, COMAL!

This article series may have an extremely short lifespan if I don't get some feedback from YOU as users of COMAL. I'll make the first two pieces a general introduction to the standard commands, their syntax, and some comparison to other languages (BASIC, Pascal, etc.). Anything beyond that type of article would require a determined body of users interested in using the language. I've been looking for a small group of fellow computerists to share in purchasing some of the literature and PD software available from the U.S. Users' group, so, let me know if you'd like to be involve with a "sub-group" like that in C.U.G.S.

On to COMAL.

First, the COMAL 0.14 system is usually placed on every disk with a fast loader and a boot program. To start COMAL, put in your disk (switch 128 to 64 mode) and LOAD"COMAL*",8. This will load the boot program which will load the full OS for COMAL when run. As the program loads you'll get some messages to read on the screen, and one requester - you'll be asked if you want to INCLUDE error messages in the program. If you choose YES (I usually do) then you lose the use of part of the sprite area (not much). If you say NO then all errors force a disk read search for the appropriate message (time consuming).

After loading the autoboot program searches for a welcome screen called "HI", which prints some messages and looks for a "MENU". If not found, your disk drive error light will flash and "HI" will end abruptly and remove itself from memory. The MENU is usually a short description of the contents of the disk, which is NOT found on every disk. You may load and alter "HI" to NOT look for MENU, as it is a simple COMAL program. If the disk you're using produced an error light, clear identify and clear the error by typing STATUS and hit <RETURN>. (Neat, huh? No more OPEN15,8,15 etc...etc.)

First, there are many commands in COMAL that resemble BASIC commands, especially the common ones we're most used to, so learning them ought to come easy. The DIFFERENCE is the functioning of the various commands - LOAD and SAVE default to the disk drive device #8, syntax of a typed direct command is checked upon pressing the <RETURN> key, the directory is available (non-destructive to the program in memory) with the command CAT.

LOAD any program with the syntax 'LOAD"<name>". LIST the program as you would in BASIC (it's a good idea to LIST a program before you RUN it - COMAL WILL NOT RUN BASIC - in fact, BASIC can "lock up" the program - but, a program that LISTS in COMAL will RUN in COMAL). Presuming you've loaded a COMAL program you may edit it using the full screen editor you've come to know and love, with the added convenience of an IMMEDIATE SYNTAX CHECK to help avoid typos! Once you're satisfied with your efforts you may RUN the program and enjoy your efforts. Programs may be interrupted with the <RUN/STOP> key. Oh, by the way, there is an autoloader/run command which does a load and a run automatically - CHAIN"<name>" would load and run <name>.

The PRINT statement is available to you (in immediate or program mode). Also available, the much desired PRINT USING. For the budding programmer, most of the "wish list" from BASIC 2.0 has been implemented, and more. AUTO produces automatic line numbering (but, remember, the line numbers in COMAL are STRICTLY FOR EDITING PURPOSES - they are not GOTO labels!). RENUM will renumber lines to the standard increment of 10's; DEL <num1-num2> will DELETE a range of lines; LIST <num1-num2> lists a range of lines.

Output defaults to the screen, but may be redirected to the (line) printer at any time with (OUTPUT) SELECT "LP:" - type (OUTPUT) SELECT "DS:" to redirect output back to the screen. Thus, to LIST a program to the printer type:

```
OUTPUT SELECT "LP:" <RETURN>
LIST <RETURN>
OUTPUT SELECT "DS:" <RETURN>
(or everything you type will continue to be
sent to the printer.)
```

Compare that sequence of events with its BASIC (2.0) equivalent:

```
OPEN4,4 : CMD4 : LIST
PRINT#4 : CLOSE4
```

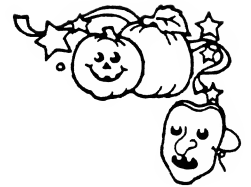
N.B. '//' (double hash marks) are the equivalent of REM; some COMAL interpretations permit you to ENTER '!' and interpret this as '//' upon LISTing. Also, the colon (:) has special significance in COMAL; thus, multiple command lines are NOT allowed. This might seem a bit awkward to long-time BASIC users, but it DOES smarten up the logic and flow of a program.

Variables may be of (virtually) any length (up to 79 characters). The first character must be alphabetic, but the rest may mix alphabetic, numeric and some special characters. All variables must be DIMensioned (the equivalent to declaring a variable in Pascal) and may be treated GLOBALLY or LOCALLY. Assignment of content is done with the assignment sign ':='. ('=' is used in equivalency tests.)

You can create arrays with text or numeric variables by proper DIMensioning.

Just a "taste" of next month's venue - because COMAL has a public domain subset of Logo's TURTLE GRAPHICS, and an extended set of graphics commands to manage the 64's sound and graphics, even beginning programmer's can create some remarkable sight and sound displays with simple, easily understood commands. For example, what do YOU think (never having used COMAL or LOGO) the following regular C64 COMAL commands mean:

```
DEFINE
SETGRAPHIC
BACKGROUND
BORDER
FILL
HIDESPRITE
PRIORITY
```



These are only a few of the commands available to allow incredibly easy graphic manipulation.

As we'll see next month!

YOU CAN USE A BBS!

CAN YOU USE A BBS?

(Reprinted from "Printips"
Campbell Printing Co.
Rensselaer, IN)

BBS stands for Bulletin Board System. It's a computerized version of the cork and push-pin bulletin boards we've all seen ... sort of an electronic information booth. Virtually anyone with a computer and a telephone can start a bulletin board system, and there are now over 4,000 in operation throughout the country. That number is uncertain because the equipment necessary for setting up a BBS is constantly decreasing. In essence, starting up a BBS is as simple as hooking up a telephone answering machine.

This month, we'll explore the whys and wherefores of the electronic bulletin board. The topic may seem a little strange but don't make the mistake of thinking that electronic bulletin boards are only used by the computer enthusiast. These systems have already begun to pass into the business mainstream. And the more your business relies on computers, the greater potential benefits of setting up a BBS.

RECIPE FOR A BULLETIN BOARD

Remember, the computer is an appliance, albeit an incredibly flexible one, and the BBS is just another of its possible uses. An electronic bulletin board consists of three key elements: the machine itself (hardware), a BBS program (software), and a human.

On the hardware end, you'll need a computer, a telephone line, and a device called a MODEM (an acronym, sort of, for MODulator/DEModulator). The modem translates computer data over a phone line. A modem at the other end of the line receives the sounds and converts them back into signals that make sense to the computer. Ideally, the computer should have plenty of storage space. And to help eliminate confusion, the BBS phone line should be used only by the computer.

You need more than hardware, however. You'll also need a computer program that 'tells' the hardware how to be a bulletin board. Over 40 BBS software packages are available to choose from, and about a dozen are placed into the public domain by the authors ... they're available for free.

The third ingredient for a successful BBS is the Sysop, or SYStem OPERator. This is a human who is responsible for the maintenance of the bulletin board. Routine maintenance entails things like deleting old messages, responding to questions or comments from callers, typing in notices of special events or items of note, and even occasionally modifying the way the BBS is organized.

The human element is, of course, the most important. The sysop should be someone who is comfortable with computers, but not necessarily an expert. The sysop must take the time to set up the BBS and understand how to run it. These are not difficult requirements, but they must be appreciated.

The sysop is, after all, the BBS's human representative anyone using the board will in some way be dealing with this person.

WHO USES BULLETIN BOARDS?

Computer hobbyists in Chicago created the first BBS on a microcomputer in the late 1970's. They used the system to swap hardware and software, post club meeting times, and exchange technical tips. BBS communication has since outgrown the computer culture that spawned it. As computers become an increasingly common appliance, groups and individuals whose interests extend beyond

computers have found bulletin boards to be an efficient means of communication. A few examples:

*The Association of Individual Investors in Chicago uses bulletin boards for members to swap ideas, and clients of a discount brokerage firm in New York can call a BBS to get quotes or place orders.

*Health professionals in other states use bulletin boards as information clearinghouses. Doctors at a Cleveland hospital even offered electronic house calls through a BBS they called "Doc in the Box".

*Many other professionals, such as lawyers, social workers, broadcasters, and architects have their own boards. A printer in Wisconsin actually uses a BBS so that customers with computers can send the text they want printed. It saves time because the job does not have to be keyed-in from paper. And by saving time, this BBS also saves money both the printer's and the customer's. Many magazines also use BBS's to get instant feedback from readers.

NEW CUGS DISK LIBRARY ADDITIONS:

COMMUNICATIONS 7 XG

SPRINT IV	DISPLAY AND PRINT SEQ FILES
LASER 2.2	TERMINAL FOR 1650 AT 300 BD
TELE-CHESS 1.4	PLAY CHESS VIA MODEM
LYNX XII	COMB. MULTIFILED PGMS INTO 1
LIBRARY 9.0	" " " "
CESAR'S TERM.	MULTICHOICE EVERYTHING

COMMUNICATIONS 9 XI

SPRINT IV	DISPLAY AND PRINT SEQ FILES
TOUCHZOOM	PUNTER PROTOCOL TERMINAL PG
TERMINAL174BOOT	CHOOSE PROTOCOL/BAUD

SOUND 13 SM

SIDPIC V2.5	COMPUTE'S SID PLAYER
MUSIC.MENU	RELIGIOUS MUSIC
COLLECTION 1	MUSIC WITH POEMS
SPEECH	PRESS A KEY AND COMPUTER TALKS
STAFF WRITER	PRINTS A MUSIC STAFF
IMPROVISOR	16 SHORT PIECES OF MUSIC

COMAL SECTION:

N.B. All COMAL disks carry the complete COMAL system, so only PROGRAMS and text files are listed

0.14 COMAL 1 LA

INFORMATION83NOV	S
INSTRUCTIONS0.14	S
>-----<	S
>COMAL PROGRAMS<	S
>-----<	S
SEE'INFORMATION	P
SEE'INSTRUCTIONS	P
LOGO'BOOK'SAMPLE	P
SNOWFLAKE	P
SPRITE/TURTLE	P
LOGO EMULATOR	P
LANDER	P
MUSIC	P
SANTA'GAME	P
BOUNCE	P
SPRITE'DESIGNER	P
READ'DIRECTORY	P
PRINT'DIRECTORY	P
EXPRESSION	P
UTILITIES	P
RECURSIONS	P
FORMATTER	P

FILE'TO'PRINT	P
FILE'TO'SCREEN	P
DISK COMMANDS	P
C64 COMAL INFO	P
REMOVE COMMENTS	P
SEE'ROLL/DEMO	P
SEE'PAGE/DEMO	P
CURSOR/DEMO	P
VALUE/DEMO	P
SHIFT/DEMO	P
JIFFY/DEMO	P
QUICKSORT/DEMO	P
JOYSTICK/DEMO	P
PADDLE/DEMO	P
DISK'GET/DEMO	P
LOGICAL'OPS/DEMO	P
SANTA	S
>-----<	S
>END OF COMAL	< S
>DEMO PROGRAMS	< S
>-----<	S

0.14 COMAL 2 LB

>-----<	S
>COMAL PROGRAMS<	S
>-----<	S
MISSING.DAT	S
PHONE.DAT	S
COMAL AND YOU	S
CURVES	S
DESIGN	S
KENO GAME	S
MISSING LETTERS	S
PHONE LIST	S
PLAYNET INFO	S
READY AIM DRAW	S
SPRITE ANIMATION	S
TURTLE TUTOR	S
CURVES'	P
DESIGN'	P
KENO'GAME	P
MISSING'LETTERS	P
PHONE'LIST	P
PLAYNETSIMULATOR	P
READY'AIM'DRAW	P
SPRITE'ANIMATION	P
TURTLE'TUTOR	P
DIR'EDITOR.MEM	P
BOOT'DIR'EDITOR	P
DIR'MANIPULATOR	P
DIRECTORY'EDITOR	P
DISK'EDIT/PROTCT	P
DISK'EDITOR	P
COMAL ARTICLE	S
DIR EDITOR.TXT	S
DISK EDITOR.TXT	S
DISK PROTECT.TXT	S
FILE FIXING.TXT	S
HELP.DAT	S
FILE'TO'PRINT2	P

0.14 COMAL 3 LC

INFORMATION	S
>-----<	S
>ARTICLES<	S
COMAL ARTICLE	S
DATA BASE.TXT	S
FONT EDITOR.TXT	S
SIMPLE TERM.TXT	S
KELLYS BEACH.TXT	S

>---DATA FILES---< S

>DO NOT LOAD<	S
BEACHNAMES.DAT	S
BEACHSPRITES.DAT	S
BEACHVERBS.DAT	S
FFDB.DAT	S
OBJECTTABLE.DAT	S
VERBTABLE.DAT	S
>---COMAL---<	S
>---PROGRAMS---<	S
SET.ART DECO.B	P
SET.STANDARD.B	P
FONT'EDITOR	P
FREE'FORM'DB	P
KELLYS'BEACH	P
CRG14.COMPACTOR	P
CRG14.FILEWRITER	P
CRG14.VIEWER	P
SIMPLE'TERM	P
>>---ARTICLES---<<	S
DIRECTORY	S
BBS.CRG	S
COMPACT PIX.TXT	S
CALVIN.CRG	S
COMPACT PIX	S
GOOFYS CAR.CRG	S
LOON.CRG	S
WDLOGO	P

0.14 COMAL 4 LD

>-----<	S
>---ARTICLES---<	S
>-----<	S
DAT.FFDB	S
DATA BASE.TXT	S
PRINT DIR.TXT	S
DISK EDITOR.TXT	S
DIR PROBE.TXT	S
NEW RECOVERY.TXT	S
EDITOR-PART2.TXT	S
>-----<	S
>---COMAL---<	S
>---PROGRAMS---<	S
>-----<	S
FREE'FORM'DB	P
DIR'PROBE'BOOT	P
DIRECTORY'PROBE	P
DISK'EDITOR'BOOT	P
DISK'EDITOR	P
PRINT'DIRECTORY	P
BASIC2COMAL.TXT	S
EDIT BOX.TXT	S
FONT EDITOR.TXT	S
STAR TREK.TXT	S

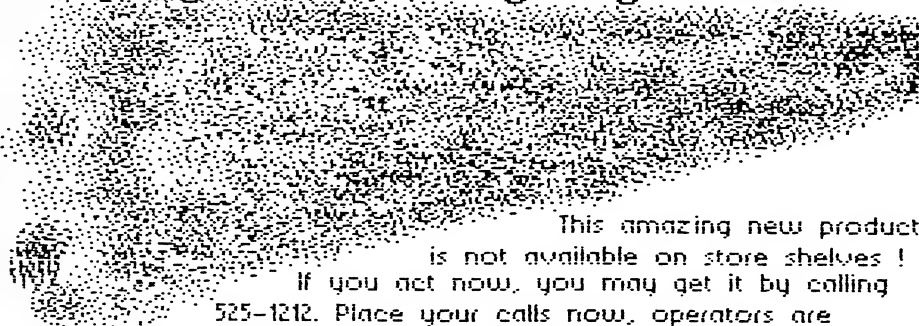
SET.ART DECO.B	P
SET.STANDARD.B	P
BASIC2COMAL-P1	P
BASIC2COMAL-P2	P
EDIT'BOX	P
FONT'EDITOR	P
STAR'TREK'DB	P
RAN.STARTREK	R

0.14 COMAL 5 LE

BOOT C64 COMAL	P
>-----<	U
>-----<	U
C64 COMAL 0.14	P
COMALERRORS	S
ML.FASTLOAD-MSD	P
ML.SIZZLE	P
HI	P
MENU	P
NAMES.DAT	S
>-----<	U
< COMAL 0.14 <	U
< PROGRAMS <	U
>-----<	U
EDIT'WHEEL'DATA	P
GENERIC'DB	P
PRINT'SHOWS	P
WHEEL'OF'FORTUNE	P
>-----<	U
< TEXT FILES <	U
>-----<	U
COMAL ARTICLE	S
DOCTOR WHO.TXT	S
EDIT RANDOM.TXT	S
INFO.TXT	S
STAR TREK.TXT	S
WHEEL OF \$.TXT	S
>-----<	U
< DATA FILES <	U
>-----<	U
HRG.WHEEL	S
LOAD/SAVE.MEM	P
RAN.DOCTORWHO	R
RAN.STARTREK	R
RAN.WHEEL	R

The Low Bytes... **MORE MIRTH!**

New Product of the Week Just off the factory line of J&G
Introducing the new "Debugger". Rids
your program of all bugs. Big or small!

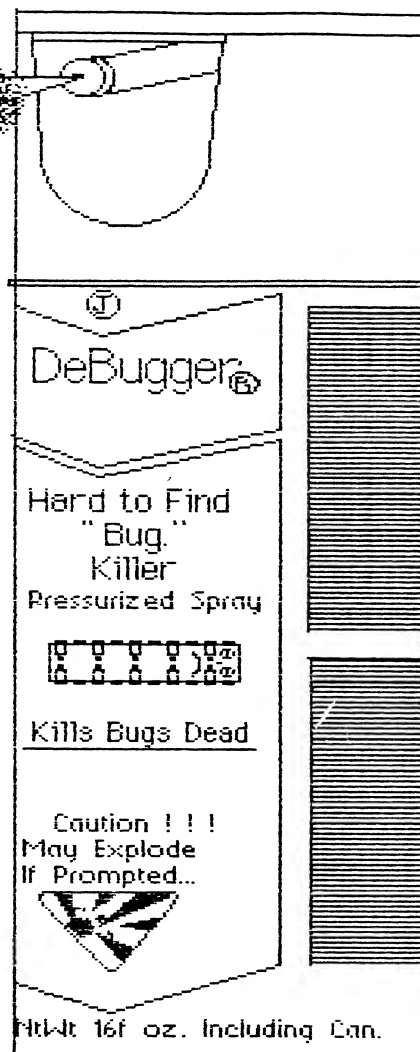


This amazing new product
is not available on store shelves !
If you act now, you may get it by calling
525-1212. Place your calls now, operators are
standing by. For just \$119.00 plus tax you can rid
your programs of that annoying BUG by pushing a button
and spraying it at a distance of 8" or more. CALL NOW !!!

Quantities are Limited

<input type="radio"/> 10 REM *****	<input type="radio"/>
<input type="radio"/> 20 REM * Capute's Guzzett Aug/88 *	<input type="radio"/>
<input type="radio"/> 30 REM * Sprite Designer #37 *	<input type="radio"/>
<input type="radio"/> 40 REM * Please read Aug issue *	<input type="radio"/>
<input type="radio"/> 50 REM * #49 for instructions *	<input type="radio"/>
<input type="radio"/> 60 Rem *****	<input type="radio"/>
<input type="radio"/> 70 FOR X= 1 to 40	<input type="radio"/>
<input type="radio"/> 80 FOR Y= 1 to 25	<input type="radio"/>
<input type="radio"/> 90 READ A : POKE A,X+Y	<input type="radio"/>
<input type="radio"/> 100 POKE X,PEEK Y	<input type="radio"/>
<input type="radio"/> 110 NEXT X	<input type="radio"/>
<input type="radio"/> 120 NEXT Y	<input type="radio"/>
<input type="radio"/> 130 PRINT "Caputes Sprite designer #37"	<input type="radio"/>
<input type="radio"/> 140 PRINT "By John Didit Again"	<input type="radio"/>
<input type="radio"/> 150 SPRDEF	<input type="radio"/>
<input type="radio"/> 160 PRINT "Please read instruction	<input type="radio"/>
manual if you don't know how"	<input type="radio"/>
<input type="radio"/> 170 DDINT " to do it "	<input type="radio"/>

WARNING: The surgeon general has found that "Computing is Fun"



B. Bircher 8/88

ACHTUNG!

ALLES LOOKENPEEPERS ACHTUNG!!

Das compütenmaschine is nicht für gefingerpoken und mittengrabben,
Ist easlich schnappen der springwerk, blowenfösen, und poppencorken
mit spittzen-spärken!! Ist nicht für gewerken by das dummköpfen!

Das rubbernecken zeitseeren keepen handes in das pocket -
relaxen und vatch das blinken-lights!

ATTACH THIS TO YOUR COMPUTER - ESPECIALLY AT THE OFFICE!